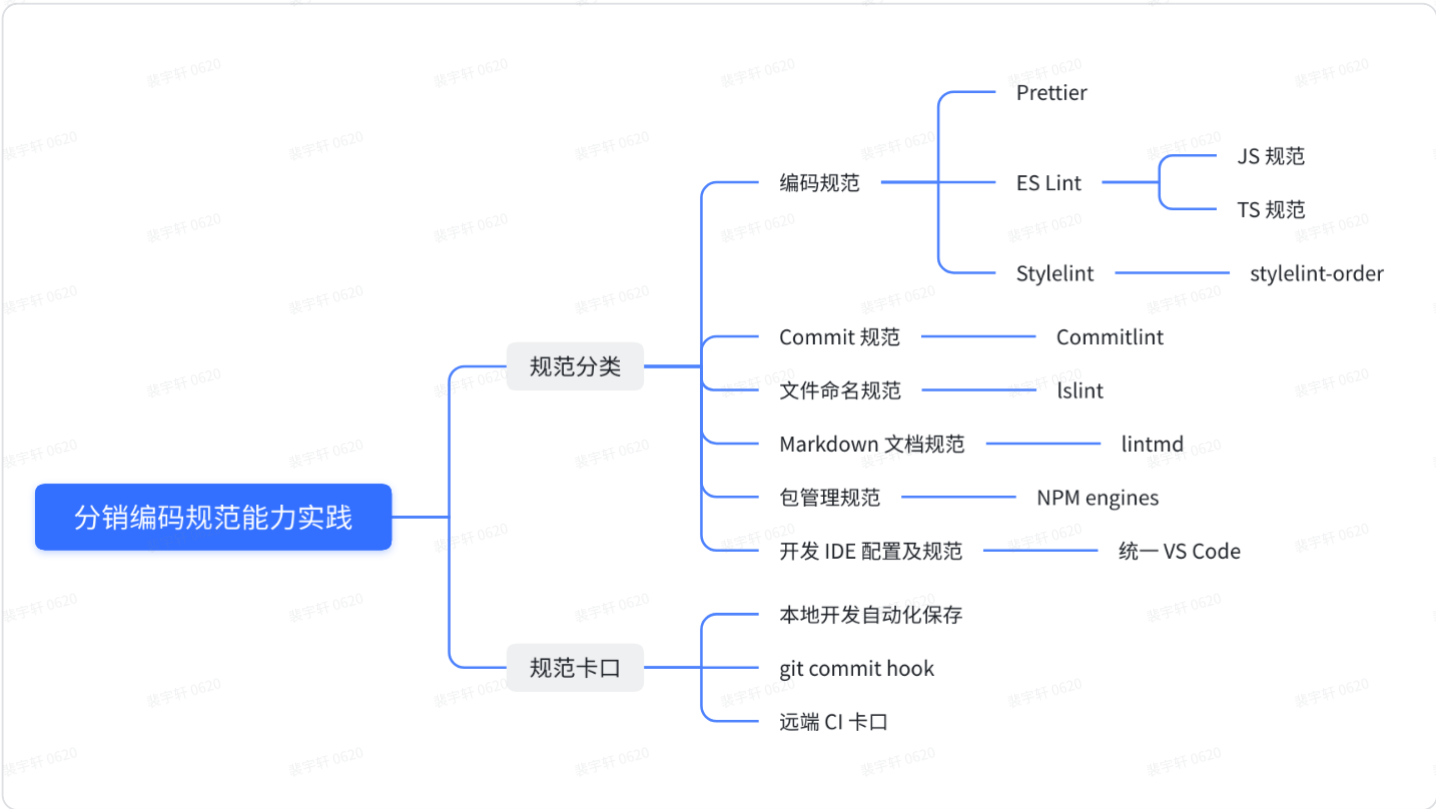


前端项目通用规范实践

概览

付费小说团队的项目基本都是 Web 类项目，所以规范实践基本只覆盖了 Web 相关的场景



实践原则

- 安全性：优先规避问题，其次探寻开发舒适度
- 统一性：强制统一，个人最优需向团队最优妥协
- 自动化：力争通过"自动化+机器"的方式解决问题，即对规范完全无感知也能“按规范”开发
- 通用性：覆盖现有项目所有能覆盖的必要场景
- 从严性：所有规范统一向最佳实践看齐，特殊场景需要豁免统一走行内 disable 形式，并在 Code Review 阶段严格把控
- 各lint设计原则：继承官方/业内推荐 + 自定义添加/修改。即高优参考/继承业内优秀实践，并根据实际情况进行补充和修改，避免重复造轮子
- 适合项目的规范 + 必要的自动化卡口 = 安全性 ↑ + 开发效率 ↑
- 新项目严格按照规范，老项目根据实际情况进行逐步迁移

实践细节

规范分类

编码规范

ESLint

目标：规范项目中的 js/jsx/ts/tsx 文件

主要面向对象：**TS + React项目**

要点：

- 配置文件类型选择：
json：推荐json配置文件而不是js文件，书写更规范（避免出现key带引号和不带引号）；
eslint对json文件的解读更好，hover可直接展示配置说明
js：动态化配置能力更强，可扩展性更强
- 所有项目都会使用 ESLint 来完成对 TS/JS 逻辑的规范性校验，这里需要特别关注的是，当项目中同时存在 JS 以及 TS 文件时，我们需要通过 ESLint 配置的 **overrides** 能力来同时兼容两种文件的校验能力，仅使用一种规范通常会出现许多问题

相关包说明：

- eslint**：ESLint的核心代码
- @typescript-eslint/parser**：ESLint解析器，用于解析typescript，从而检查和规范Typescript代码
- @typescript-eslint/eslint-plugin**：ESLint插件，包含了各类定义好的检测Typescript代码的规范，对应下文配置的"plugin:@typescript-eslint/recommended"
- eslint-plugin-react-hooks**：ESLint rules for React Hooks，与react官方保持一致，对应下文配置的"plugin:react-hooks/recommended"
- eslint-config-prettier**：解决ESLint中的样式规范和prettier中样式规范的冲突，以prettier的样式规范为准，使ESLint中的样式规范自动失效。**对应eslintrc配置extends-"prettier"**
- eslint-plugin-prettier**：将prettier样式规范作为ESLint规范来使用，同样将格式问题以error的形式抛出，即rule-"prettier/prettier"，可在rules-"prettier/prettier"中进行自定义配置。**对应eslintrc配置plugin-"prettier"**
- 补充说明：extends-"prettier"（eslint-config-prettier）+ plugin-"prettier"（eslint-plugin-prettier）= extends-"prettier:prettier-recommended"，详细说明见以下链接：
 - <https://github.com/prettier/eslint-config-prettier/blob/main/CHANGELOG.md#version-800-2021-02-21>

- <https://github.com/prettier/eslint-plugin-prettier#recommended-configuration>

配置相关包下载:

```
1 yarn add eslint prettier @typescript-eslint/parser @typescript-eslint/eslint-plugin
  eslint-plugin-react-hooks eslint-config-prettier eslint-plugin-prettier --dev
```

参考配置如下:

```
1 // .eslintrc.json
2
3 {
4   "parser": "@typescript-eslint/parser",
5   "extends": [
6     // ts官方推荐
7     'plugin:@typescript-eslint/recommended',
8     // react-hooks
9     'plugin:react-hooks/recommended',
10    // eslint-config-prettier + eslint-plugin-prettier
11    https://github.com/prettier/eslint-plugin-prettier#recommended-configuration
12
13    简化写法: eslint-config-prettier + eslint-plugin-prettier
14    'plugin:prettier/recommended',
15  ],
16  "plugins": ["@typescript-eslint"],
17  "env": {
18    "browser": true,
19    "es6": true,
20    "node": true
21  },
22  "rules": {
23    // 命名规范
24    "@typescript-eslint/naming-convention": [
25      1,
26      {
27        "selector": "interface",
28        "format": ["PascalCase"],
29        "prefix": ["I"]
30      },
31      {
32        "selector": "enum",
33        "format": ["PascalCase"],
34        "prefix": ["E"]
35      }
36    ]
37  }
38 }
```

```

35     },
36     {
37       "selector": "typeAlias",
38       "format": ["PascalCase"],
39       "prefix": ["T"]
40     },
41     {
42       "selector": "enumMember",
43       "format": ["PascalCase"]
44     },
45     {
46       "selector": "function",
47       "format": ["camelCase"],
48       "custom": {
49         "match": true,
50         "regex": "^[handle|get|render|request|create|retrieve|search|delete]"
51       }
52     }
53   ]
54 },
55
56 // js相关
57 "overrides": [
58   {
59     "files": ["*.js", "*.jsx"],
60     "extends": ["eslint:recommended", "plugin:prettier/recommended"]
61   }
62 ]
63 }

```

常见问题及解决方案：

- eslint不生效：大多数都是包安装问题/规则配置写法问题，可通过eslint本地服务查看报错情况
- eslint服务提示找不到eslint-config-prettier等包：清楚本地node_modules，重启项目并重新下载依赖



mattymess commented on 22 Jan

@jamesgordo I was able to get it working the other day. I just wiped out `node_modules` and did a fresh `npm i`. It still wasn't working, but after I restarted VSCode, it did 🙌



- prettier格式化后还标红提示"prettier/prettier"：重启项目

- eslint与prettier规则打架：比如在配置prettier自动格式化后，还是会报prettier/prettier的缩进错误（eslint缩进判断"?"有效，prettier判断其无效）。在配置prettier格式化背景下，应删掉eslint-rules中对相关样式的自定义，以避免与prettier/prettier的冲突

Prettier

背景：ESLint 本身就已经具备了很完善的代码检测能力，但是我们还可以借助 prettier 补齐 ESLint 无法覆盖的例如函数换行等格式化能力，进一步提升规范性，而且比起 ESLint，Prettier 还可以覆盖 HTML、CSS 等更多类型的文件格式化

相关包：

- [prettier](#)

配置相关包下载：（配置eslint时下载后此处不再需下载）

```
1 yarn add prettier --dev
```

参考配置如下：

```
1 // .prettierrc.js
2
3 module.exports = {
4   arrowParens: 'avoid',
5   bracketSpacing: true,
6   printWidth: 120,
7   semi: true,
8   singleQuote: true,
9   tabWidth: 4,
10  trailingComma: 'all',
11  useTabs: false,
12 };
```

Stylelint

目标：规范项目中的 css/less 等样式文件

相关包：

- [postcss-less](#)：less解析包
- [stylelint](#)

- **stylelint-order**: stylelint中与属性顺序有关的插件
- **stylelint-config-standard**: extends `stylelint-config-recommended`，额外支持一些提案中的样式规范

包下载:

```
1 yarn add postcss-less stylelint stylelint-order stylelint-config-standard --dev
```

要点:

- class类名规范 (待定): 不采用业内常见的中划线, 采用"小驼峰+下划线", 在满足模块划分的基础上支持双击选取, 提升开发效率
- 属性顺序: 规范项目中的 CSS 属性顺序 (社区通用实践: 布局、盒模型、文字排版、装饰颜色、内容), 具体通过 **"order/properties-order"**

参考配置如下:

```
1 {
2   "extends": "stylelint-config-standard",
3   "plugins": ["stylelint-order"],
4   "overrides": [
5     {
6       "files": ["src/**/*.less"],
7       "customSyntax": "postcss-less",
8     }],
9   "rules": {
10    "indentation": 4,
11    "max-nesting-depth": 4,
12    // class命名规范: camelCase(_camelCase)+
13    "selector-class-pattern": "^[a-z][a-zA-Z0-9]*(_[a-z][a-zA-Z0-9]+)*$",
14    "order/order": [
15      "declarations",
16      "custom-properties",
17      "dollar-variables",
18      "rules",
19      "at-rules"
20    ],
21    "order/properties-order": [
22      "position",
23      "top",
24      "right",
25      "bottom",
26      "left",
```

```
27     "float",
28     "clear",
29     "display",
30     "flex",
31     "flex-grow",
32     "flex-shrink",
33     "flex-basis",
34     "flex-flow",
35     "flex-direction",
36     "flex-wrap",
37     "justify-content",
38     "align-content",
39     "align-items",
40     "align-self",
41     "order",
42     "grid",
43     "grid-template-rows",
44     "grid-template-columns",
45     "grid-template-areas",
46     "grid-auto-rows",
47     "grid-auto-columns",
48     "grid-auto-flow",
49     "grid-column-gap",
50     "grid-row-gap",
51     "grid-template",
52     "grid-template-rows",
53     "grid-template-columns",
54     "grid-template-areas",
55     "grid-gap",
56     "grid-row-gap",
57     "grid-column-gap",
58     "grid-area",
59     "grid-row-start",
60     "grid-row-end",
61     "grid-column-start",
62     "grid-column-end",
63     "grid-column",
64     "grid-column-start",
65     "grid-column-end",
66     "grid-row",
67     "grid-row-start",
68     "grid-row-end",
69     "table-layout",
70     "empty-cells",
71     "caption-side",
72     "border-collapse",
73     "border-spacing",
```

```
74 "list-style",
75 "list-style-type",
76 "list-style-position",
77 "list-style-image",
78 "ruby-align",
79 "ruby-merge",
80 "ruby-position",
81 "box-sizing",
82 "width",
83 "min-width",
84 "max-width",
85 "height",
86 "min-height",
87 "max-height",
88 "padding",
89 "padding-top",
90 "padding-right",
91 "padding-bottom",
92 "padding-left",
93 "border",
94 "border-width",
95 "border-top-width",
96 "border-right-width",
97 "border-bottom-width",
98 "border-left-width",
99 "border-style",
100 "border-top-style",
101 "border-right-style",
102 "border-bottom-style",
103 "border-left-style",
104 "border-color",
105 "border-top-color",
106 "border-right-color",
107 "border-bottom-color",
108 "border-left-color",
109 "border-image",
110 "border-image-source",
111 "border-image-slice",
112 "border-image-width",
113 "border-image-outset",
114 "border-image-repeat",
115 "border-top",
116 "border-top-width",
117 "border-top-style",
118 "border-top-color",
119 "border-top",
120 "border-right-width",
```


121 "border-right-style",
122 "border-right-color",
123 "border-bottom",
124 "border-bottom-width",
125 "border-bottom-style",
126 "border-bottom-color",
127 "border-left",
128 "border-left-width",
129 "border-left-style",
130 "border-left-color",
131 "border-radius",
132 "border-top-right-radius",
133 "border-bottom-right-radius",
134 "border-bottom-left-radius",
135 "border-top-left-radius",
136 "outline",
137 "outline-width",
138 "outline-color",
139 "outline-style",
140 "outline-offset",
141 "margin",
142 "margin-top",
143 "margin-right",
144 "margin-bottom",
145 "margin-left",
146 "color",
147 "background",
148 "background-image",
149 "background-position",
150 "background-size",
151 "background-repeat",
152 "background-origin",
153 "background-clip",
154 "background-attachment",
155 "background-color",
156 "background-blend-mode",
157 "isolation",
158 "clip-path",
159 "mask",
160 "mask-image",
161 "mask-mode",
162 "mask-position",
163 "mask-size",
164 "mask-repeat",
165 "mask-origin",
166 "mask-clip",
167 "mask-composite",

```
168     "mask-type",
169     "filter",
170     "box-shadow",
171     "opacity",
172     "visibility",
173     "overflow",
174     "overflow-x",
175     "overflow-y",
176     "vertical-align",
177     "columns",
178     "columns-width",
179     "columns-count",
180     "column-rule",
181     "column-rule-width",
182     "column-rule-style",
183     "column-rule-color",
184     "column-fill",
185     "column-span",
186     "column-gap",
187     "orphans",
188     "writing-mode",
189     "text-combine-upright",
190     "unicode-bidi",
191     "text-orientation",
192     "direction",
193     "text-rendering",
194     "font-feature-settings",
195     "font-language-override",
196     "font",
197     "font-style",
198     "font-variant",
199     "font-weight",
200     "font-stretch",
201     "font-size",
202     "font-family",
203     "line-height",
204     "text-overflow",
205     "white-space",
206     "overflow-wrap",
207     "word-wrap",
208     "word-break",
209     "line-break",
210     "hyphens",
211     "text-align",
212     "text-align-last",
213     "text-justify",
214     "font-synthesis",
```

215 "font-size-adjust",
216 "letter-spacing",
217 "font-kerning",
218 "word-spacing",
219 "text-transform",
220 "quotes",
221 "tab-size",
222 "text-indent",
223 "text-emphasis",
224 "text-emphasis-style",
225 "text-emphasis-color",
226 "text-emphasis-position",
227 "text-decoration",
228 "text-decoration-color",
229 "text-decoration-style",
230 "text-decoration-line",
231 "text-underline-position",
232 "text-shadow",
233 "image-rendering",
234 "image-orientation",
235 "image-resolution",
236 "shape-image-threshold",
237 "shape-outside",
238 "shape-margin",
239 "transform-style",
240 "transform",
241 "transform-box",
242 "transform-origin",
243 "perspective",
244 "perspective-origin",
245 "backface-visibility",
246 "transition",
247 "transition-property",
248 "transition-duration",
249 "transition-timing-function",
250 "transition-delay",
251 "animation",
252 "animation-name",
253 "animation-duration",
254 "animation-timing-function",
255 "animation-delay",
256 "animation-iteration-count",
257 "animation-direction",
258 "animation-fill-mode",
259 "animation-play-state",
260 "scroll-behavior",
261 "scroll-snap-type",

```
262     "scroll-snap-destination",
263     "scroll-snap-coordinate",
264     "resize",
265     "cursor",
266     "touch-action",
267     "caret-color",
268     "ime-mode",
269     "object-fit",
270     "object-position",
271     "content",
272     "counter-reset",
273     "counter-increment",
274     "will-change",
275     "pointer-events",
276     "z-index",
277     "all",
278     "page-break-before",
279     "page-break-after",
280     "page-break-inside",
281     "widows"
282 ],
283 "selector-pseudo-class-no-unknown": [
284   true,
285   {
286     "ignorePseudoClasses": ["global"]
287   }
288 ],
289 }
290 }
291
```

commit 规范

目标：规范 commit message

相关包：

- [commitlint](#)

包下载：

```
1 yarn add @commitlint/cli @commitlint/config-conventional --dev
```

参考配置见：

```
1 // commitlint.config.js
2
3 module.exports = {
4   extends: ['@commitlint/config-conventional']
5 };
```

文件/文件夹命名规范

背景：统一文件夹命名和各类文件命名，一方面是提高规范性，另一方面为避免在各种场景中产生因为大小写产生的疑难杂症（如不同系统对大小写敏感度不同、Git 提交时对大小写不敏感）

相关包：

- [ls-lint](#)

原则：

- 目录（文件夹）使用中划线，文件名使用小驼峰 / 中划线 / 点分割
- 有特殊需求的文件可通过限制文件目录进行补充覆盖

```
1 # .ls-lint.yml
2 ls:
3   src:
4     .dir: kebab-case
5     .js: snake_case | kebab-case | pointcase
6     .jsx: snake_case | kebab-case | pointcase
7     .ts: snake_case | kebab-case | pointcase
8     .tsx: snake_case | kebab-case | pointcase
9     .css: snake_case | kebab-case | pointcase
10    .less: snake_case | kebab-case | pointcase
11    .json: snake_case | kebab-case | pointcase
```

Markdown 文档规范

对于项目中的 markdown 文档，我们也希望能借鉴[社区上的优秀实践](#)，并通过 [lint-md](#) 工具实现自动化约束，具体配置如下：

```
1 {
2   "excludeFiles": [],
3   "rules": {
4     "no-long-code": 0,
5     "no-trailing-punctuation": 0
6   }
7 }
```

统一包管理规范

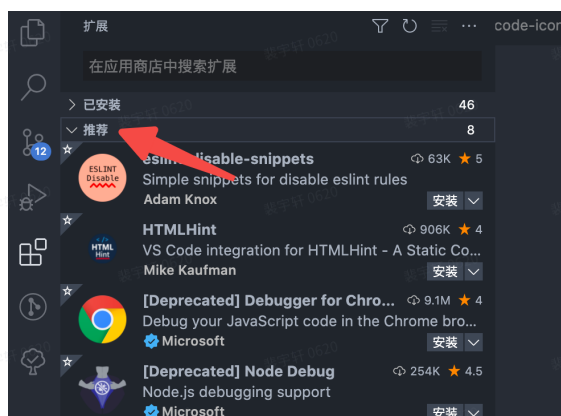
📖 付费小说包管理操作指南与规范约束

IDE配置（以vs code为例）

背景：在日常项目开发中，经常遇到的情况是，尽管项目中配置了各类严格的规范和自动化卡口，但上手开发时还是会有不同的报错和问题，**常需要补充下载IDE插件或修改配置来解决**，且不同项目中IDE的配置要求不同，所以**规范项目对应的IDE配置**也是非常重要的，消除不同同学在同一项目的IDE配置的gap，实现无障碍上手项目，降低协作开发成本，提高开发效率

插件配置

配置文件说明（.vscode/extensions.json）：vs code插件汇总，可快速安装vs code基本插件和项目所需插件。建议手动按个安装，以避免重复安装相似功能插件，造成功能冲突 / IDE变卡顿



推荐插件说明见：[📖 vscode好用插件汇总](#)

推荐配置见：

```
1 // .vscode/extensions.json
2
3 {
```

```

4   "recommendations": [
5       "formulahendry.auto-close-tag",
6       "steoates.autoimport",
7       "formulahendry.auto-rename-tag",
8       "coenraads.bracket-pair-colorizer-2",
9       "ms-ceintl.vscode-language-pack-zh-hans",
10      "streetsidesoftware.code-spell-checker",
11      "oouo-diogo-perdigao.docthis",
12      "dbaeumer.vscode-eslint",
13      "maggie.eslint-rules-zh-plugin",
14      "donjayamanne.githistory",
15      "felipecaputo.git-project-manager",
16      "eamodio.gitlens",
17      "wix.vscode-import-cost",
18      "eg2.vscode-npm-script",
19      "vunguyentuan.vscode-postcss",
20      "esbenp.prettier-vscode",
21      "iceworks-team.iceworks-style-helper",
22      "jasonnutter.search-node-modules",
23      "burkeholland.simple-react-snippets",
24      "stylelint.vscode-stylelint",
25      "gruntfuggly.todo-tree",
26      "pflannery.vscode-versionlens",
27      "vscode-icons-team.vscode-icons"
28  ]
29  }

```

工作区配置

配置文件说明（.vscode/settings.json）：vs code工作区配置，可覆盖本地用户配置，完成项目所需配置，内也含插件配置，比如Prettier、cSpell等



推荐配置见：

```

1  // .vscode/settings.json
2
3  {

```

```
// 文件相关
```

```
"files.eol": "\n",  
"files.trimTrailingWhitespace": true,  
"files.insertFinalNewline": true,  
"files.associations": {
```

```
// json + comment
```

```
".code-workspace": "jsonc",  
".babelrc": "jsonc",  
".eslintrc": "jsonc",  
".eslintrc*.json": "jsonc",  
".stylelintrc": "jsonc",  
".stylelintrc": "jsonc",  
".htmlhintrc": "jsonc",  
"htmlhintrc": "jsonc",  
"Procfile*": "shellscript",  
"README": "markdown",
```

```
/**
```

cssnext: 使用css新特性, 其特性比less迭代更新更快, 相比less仅不支持一些for等生成类语法, 详情见<https://cssnext.github.io/>

插件: 特性转换插件为postcss-preset-env, Jupyter的procss中已内置
<https://jupyter.goofy.app/api/config/tools/postcss%E9%BB%98%E8%AE%A4%E5%80%BC>

配置说明: 使用cssnext需要配置css文件为postcss文件类型, 否则开发环境下使用新特性会报错, 如父级选择器&

```
*/
```

```
"*.css": "postcss"  
},  
"files.exclude": {  
  "**/*/adapters/**/index.ts": true,  
  "**/*/adapters/**/index.js": true  
},
```

```
// 搜索相关
```

```
"search.useIgnoreFiles": true,  
"search.exclude": {  
  "**/build": true,  
  "**/output": true,  
  "**/dist": true,  
  "**/yarn.lock": true,  
  "**/package-lock.json": true,  
  "**/*.log": true,  
  "**/*.pid": true,  
  "**/.git": true,  
  "**/node_modules": true  
},
```

```
// editor
```

```
"editor.tabSize": 4, // 缩进宽度, 配合prettier缩进
```



```
48 "editor.defaultFormatter": "esbenp.prettier-vscode", // 默认prettier格式化
49 "editor.formatOnSave": true,
50
51 // 文件格式化/lint配置
52 // 保存后的行为分两步：第一步通过editor.codeActionsOnSave调用source.fixAll进行修
    复，第二步通过editor.defaultFormatter进行格式化
53 "[javascript]": {
54     "editor.defaultFormatter": "esbenp.prettier-vscode",
55     "editor.codeActionsOnSave": {
56         "source.fixAll.eslint": true
57     }
58 },
59 "[javascriptreact]": {
60     "editor.defaultFormatter": "esbenp.prettier-vscode",
61     "editor.codeActionsOnSave": {
62         "source.fixAll.eslint": true
63     }
64 },
65 "[typescript]": {
66     "editor.defaultFormatter": "esbenp.prettier-vscode",
67     "editor.codeActionsOnSave": {
68         "source.fixAll.eslint": true
69     }
70 },
71 "[typescriptreact]": {
72     "editor.defaultFormatter": "esbenp.prettier-vscode",
73     "editor.formatOnSave": true,
74     "editor.codeActionsOnSave": {
75         "source.fixAll.eslint": true
76     }
77 },
78 "[css]": {
79     "editor.defaultFormatter": "esbenp.prettier-vscode",
80     // 单独开启formatOnSave无效
81     // "editor.formatOnSave": true
82     "editor.codeActionsOnSave": {
83         "source.fixAll.stylelint": true
84     }
85 },
86 "[postcss]": {
87     "editor.defaultFormatter": "esbenp.prettier-vscode",
88     "editor.codeActionsOnSave": {
89         "source.fixAll.stylelint": true
90     }
91 },
92 "[less]": {
93     "editor.defaultFormatter": "esbenp.prettier-vscode",
```

```
94     "editor.codeActionsOnSave": {
95         "source.fixAll.stylelint": true
96     }
97 },
98
99 // eslint相关
100 // eslint失效主要原因: 所添加的eslint规则无效/错误, 可查看eslint日志
101 "eslint.packageManager": "yarn",
102 "eslint.format.enable": true,
103 "eslint.alwaysShowStatus": true,
104 "eslint.probe": ["javascript", "javascriptreact", "typescript",
"typescriptreact"],
105
106 "typescript.tsdk": "node_modules/typescript/lib",
107
108 // 插件stylelint配置
109 "stylelint.enable": true,
110 "stylelint.snippet": ["css", "less", "postcss"],
111 "stylelint.validate": ["css", "less", "postcss"],
112
113 // 插件todoTree配置
114 "todo-tree.general.tags": ["TODO:", "FIXME:"],
115 "todo-tree.highlights.defaultHighlight": {
116     "gutterIcon": true
117 },
118 "todo-tree.highlights.customHighlight": {
119     "TODO:": {
120         "foreground": "#fff",
121         "background": "#ffbd2a",
122         "iconColour": "#ffbd2a"
123     },
124     "FIXME:": {
125         "foreground": "#fff",
126         "background": "#f06292",
127         "icon": "flame",
128         "iconColour": "#f06292"
129     }
130 },
131
132 // 插件cSpell配置
133 "cSpell.words": [
134     "byted",
135     "Coppa",
136     "ecpm",
137     "GAID",
138     "geos",
139     "IDFA",
```

```
140     "IMEI",
141     "immer",
142     "maskrule",
143     "OAID",
144     "okee",
145     "Optimised",
146     "overscan",
147     "pangle",
148     "Sider",
149     "slardar",
150     "tweenjs",
151     "uikit",
152     "wifi",
153     "abtest"
154 ],
155
156 // 穿件iconTheme主题配置
157 "workbench.iconTheme": "vscode-icons"
158 }
```

规范卡口

完整严格的卡口是保证规范能够长期被执行的必要条件

本地开发自动化保存（无感卡口）

背景：通过IDE工作区配置，实现保存时自动格式化，进一步提升规范性，并为开发提效

参考配置见：IDE配置-工作区配置部分

git commit hook（绅士卡口）

通过使用 husky + lint-staged，可以保证在执行 git commit 过程中对项目中新增的逻辑进行规范性校验与拦截，结合上述实践在分销项目中的规范配置大体如下：

```
1 {
2   ...,
3   "husky": {
4     "hooks": {
5       "pre-commit": "lint-staged",
6       "commit-msg": "commitlint -E HUSKY_GIT_PARAMS"
```

```

7      }
8    },
9    "lint-staged": {
10      "**/*": [
11        "ls-lint"
12      ],
13      "**/*.json": [
14        "prettier --write"
15      ],
16      "**/*.{js,ts,tsx}": [
17        "prettier --write",
18        "eslint --fix"
19      ],
20      "src/**/*.{css,less}": [
21        "prettier --write",
22        "stylelint --fix"
23      ],
24      "**.{md,markdown}": [
25        "lint-md --fix"
26      ]
27    },
28  }

```

远端 CI 卡口（兜底卡口）

前两道卡口都有被恶意绕过的可能，因此项目中必须在 CI 也有最后一道卡口来保证远端的项目一定是符合规范的，并在绕过规范强推远端的问题出现时，能够第一时间被发现。

Gitlab 提供了已经封装好的 eslint CI 原子能力，我们只需要一键开启即可：[图 3. 如何在 MR 阶段使用代码分析](#)



配合我们在 **Branch Protection Rules** 开启 MR 时的 **Require all checks are passing before merging** 选项：

Update review rule

X

Branch name pattern

master

Rule settings

Protect matching branches

✓ Disables all force pushes to all matching branches and prevents them from being deleted.

✓ Set who are allowed to merge and push.

Allowed to merge: Maintainers

Allowed to push: Maintainers

✓ All checks must pass before branches can be merged into a branch that matches this rule.

Require all checks are passing before merging

✓ Request code owner review for current branch

Enabled

✓ Require code review before merging.

Use repository default rules

✓ Require linking work items. Block MR from being completed unless they have at least one linked work item.

Encourage traceability by checking for linked work items (JIRA / Meego / Rocket) on merge request.

Enable

Save Changes

这样每次提起 MR 时，Gitlab CI 侧都会自动执行 ESLint 校验，对于不合规的代码，MR 将被禁止合入

 Request to merge feat-tracker-jijiax... into feat-release-0.0.1

 All checks have passed ?

4 successful checks



 argus




 bits-checkRun




 [Codebase CI] Pipeline

Welcome!



 [Code Analysis] eslint



 No approval required.



Reviewers: No requested reviewers

 View rules and eligible members: 1 rules 0 members.

相关链接

codeLint基础介绍: <https://zhuanlan.zhihu.com/p/100427908>

eslint+prettier: <https://segmentfault.com/a/1190000019661168>

ls-lint: <https://everfind.github.io/tools/ls-lint.html#%E5%AE%89%E8%A3%85>

评审录屏

<https://bytedance.feishu.cn/minutes/obcne167lk6j7ml6879ju944>